REMARKS

Applicant respectfully requests reconsideration and allowance of the present application. Claims 1-25, 27-31, 34-35, and 37-43 are pending in this application. Claims 1, 11, 20, 28, 35 and 40 are independent claims.

Applicant's amendments and remarks after Final are appropriate under 37 C.F.R. §1.116 because they address the Office's remarks in the Final Action, and thus could not have been presented earlier. In addition, the amendments and remarks should be entered to place the case in better form for appeal.

Claim Rejection Under 35 U.S.C. § 103

Claims 1-25, 27-31, 34-35, and 37-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,336,139 to Feridun et al. (hereinafter "Feridun") in view of U.S. Patent No. 6,751,753 to Nguyen et al. (hereinafter "Nguyen"). The rejection is respectfully traversed.

Applicant has determined that the Nguyen patent may be sworn behind by submitting an affidavit or declaration in accordance with 37 CFR 1.131. However, at this time, Applicant elects to argue over the current rejection under 35 U.S.C. 103(a). Applicant retains the right to swear behind the Nguyen patent in the future and intends to do so if the 35 U.S.C. 103(a) is maintained.

I. Introduction

The present Application describes implementations that provide a flexible correlation system and/or method that allows a user to correlate multiple events and/or data. One benefit of the implementations is that a non-programmer or application developer can make use of predefined correlation functions that monitor user selected events and/or data parameters.

In one implementation, a state machine is used to simultaneously monitor different events and data. The state machine may be a class object. The class object may have a number of properties that are modifiable by a non-programmer. For example, the class object may include a string that represents the name of a process being monitored; a string that holds an integer that identifies the number of crashes of the process that will trigger an event if those crashes occur within a particular time period; and a string that holds another integer that defines the particular time period. Therefore, a non-programmer can define such a class object my simply filling-in the values for the process, the number of crashes and the particular time period.

The non-programmer can also create an updating consumer used to update the state of the state machine described in the above-paragraph. The updating consumer is also a class object. The updating consumer class object is separate from the state machine. In one example, the updating consumer updates an instance of the state machine when a crash occurs. Therefore, the updating consumer ensures that the state machine maintains an up-to-date number of crashes that has been observed by the state machine.

The foregoing discussion is not limiting of the claimed invention. Instead, the discussion was provided to enhance the Office's understanding of at least one implementation described in the present Application.

II. Rejection Discussion

Claim 1 recites:

> A computer-implemented method comprising:
> receiving a plurality of events;
> applying the plurality of events to a correlation function, wherein the correlation function is implemented as a state machine and is configured to correlate the plurality of events;
> identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine;
> applying the update consumer to the state machine in response to the identified event; and
> generating a specific event if the correlation function is satisfied by the plurality of events.

The Feridum patent describes the use of correlation rules implemented as state machines to recognize patterns of one or more events indicative of given conditions. According to Feridum, the observed events are examined to determine if one or more defined event patterns has occurred. If an event pattern is recognized, an event correlator may be used to take a defined action. (See Feridum, column 2, lines 42-62.)

The Feridum patent does not teach or suggest at least "identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine; [and] applying the update consumer to the state machine in response to the identified event", as recited in claim 1. The Office realizes that the Feridum patent doest not teach at least these limitations of claim 1 and attempts to cure this by combining Feridum with Nguyen.

The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-40, and column 6, lines 26-35 of Nguyen teaches the indicated limitations of claim 1. The Applicant disagrees for the following reasons.

Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen describe a status object that includes current information related to a state of a monitored system component. The status object receives information pertaining to the monitored system component and transitions its state if a transition condition associated with the status object is satisfied. If this occurs, some action is taken by the status object. (See generally, column 2, lines 1-14.) There is nothing in this section of the Nguyen patent that teaches or suggests the limitations from claim 1 that include: "identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine; [and] applying the update consumer to the state machine in response to the identified event."

The Nguyen patent further describes the use of a state machine class and a status object class. The status object class is updated if a transition state associated with an evaluation function linked to the state machine class is satisfied. Therefore, the status object class can be used to identify a state of the state machine class. (See generally, column 2, lines 24-40.) This teaching of the Nguyen patent does not approach the limitations of claim 1 that recite: "identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine; [and] applying the update consumer to the state machine in response to the identified event." More specifically, there is nothing in the Nguyen patent that indicates that the status object class is *subscribed* to an event, as recited in claim 1. At most, the status object class in Nguyen monitors a state of the state machine class. This is not the same as being subscribed to an event. Moreover, how can it be fairly said that the status object is *applied* to the state machine in response to the identified event? The Applicant submits that it cannot be fairly said as such. The status object class

in Nguyen simply monitors the state machine class; monitoring is not the same as something being applied.

Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-discussed limitations of claim 1. This portion of Nguyen describes the use of an action 130 designed to implement an action when a state of the state machine class changes. This teaching has nothing to do with "identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine; [and] applying the update consumer to the state machine in response to the identified event", as recited in claim 1.

In accordance with the above, Applicant submits that the Nguyen patent does not cure the deficiencies of the Feridum patent. Accordingly, at least claim 1 is not rendered obvious by the Feridum and Nguyen combination.

Claim 11 recites:

> A computer-implemented method comprising:
> receiving a plurality of events;
> receiving a plurality of data elements;
> identifying a plurality of correlation functions configured to correlate the plurality of events and the plurality of data elements, wherein each correlation function is implemented with an associated state machine, and wherein each state machine has an associated update consumer that updates the state of the associated state machine;
> applying the plurality of events and the plurality of data elements to the plurality of correlation functions; and
> generating a specific event if at least one of the plurality of correlation functions is satisfied.

The Feridum patent describes the use of correlation rules implemented as state machines to recognize patterns of one or more events indicative of given conditions. According to Feridum, the observed events are examined to determine if one or more defined event patterns has occurred. If an event pattern is

recognized, an event correlator may be used to take a defined action. (See Feridum, column 2, lines 42-62.)

The Feridum patent does not teach or suggest at least "each state machine has an associated update consumer that updates the state of the associated state machine", as recited in claim 11. The Office realizes that the Feridum patent doest not teach at least the indicated limitation of claim 11 and attempts to cure this by combining Feridum with Nguyen.

The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-40, and column 6, lines 26-35 of Nguyen teaches the indicated limitation of claim 11. The Applicant disagrees for the following reasons.

Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen describe a status object that includes current information related to a state of a monitored system component. The status object receives information pertaining to the monitored system component and transitions its state if a transition condition associated with the status object is satisfied. If this occurs, some action is taken by the status object. (See generally, column 2, lines 1-14.) There is nothing in this section of the Nguyen patent that teaches or suggests the limitation from claim 11 that includes: "each state machine has an associated update consumer that updates the state of the associated state machine."

The Nguyen patent further describes the use of a state machine class and a status object class. The status object class is updated if a transition state associated with an evaluation function linked to the state machine class is satisfied. Therefore, the status object class can be used to identify a state of the state machine class. (See generally, column 2, lines 24-40.) This teaching of the Nguyen patent does not approach the limitation of claim 11 that recites: "each

state machine has an associated update consumer that updates the state of the associated state machine." More specifically, there is nothing in the Nguyen patent that indicates that the status object class *updates the state* of the state machine class, as recited in claim 11. At most, the status object in Nguyen monitors the state machine class to determine when the state machine class updates. However, that act of monitoring in Nguyen does not teach or suggest that "each state machine has an associated update consumer that updates the state of the associated state machine", as recited in claim 11.

Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-discussed limitations of claim 11. This portion of Nguyen describes the use of an action 130 designed to implement an action when a state of the state machine class changes. This teaching has nothing to do with "each state machine has an associated update consumer that updates the state of the associated state machine", as recited in claim 11.

In accordance with the above, Applicant submits that the Nguyen patent does not cure the deficiencies of the Feridum patent. Accordingly, at least claim 11 is not rendered obvious by the Feridum and Nguyen combination.

Claim 20 recites:

> A computer-implemented method comprising:
> identifying a schema for creating state machines, the state machines to correlate at least two events;
> creating an instance of a particular state machine;
> defining transitions for the particular state machine by subscribing to at least one event; and
> applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object.

The Feridum patent describes the use of correlation rules implemented as state machines to recognize patterns of one or more events indicative of given conditions. According to Feridum, the observed events are examined to determine if one or more defined event patterns has occurred. If an event pattern is recognized, an event correlator may be used to take a defined action. (See Feridum, column 2, lines 42-62.)

The Feridum patent does not teach or suggest at least "applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object", as recited in claim 20. The Office realizes that the Feridum patent doest not teach at least this limitation of claim 20 and attempts to cure this by combining Feridum with Nguyen.

The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-40, and column 6, lines 26-35 of Nguyen teaches the indicated limitation of claim 20. The Applicant disagrees for the following reasons.

Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen describes a status object that includes current information related to a state of a monitored system component. The status object receives information pertaining to the monitored system component and transitions its state if a transition condition associated with the status object is satisfied. If this occurs, some action is taken by the status object. (See generally, column 2, lines 1-14.) There is nothing in this section of the Nguyen patent that teaches or suggests the limitation from claim 20 that includes: "applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object", as recited in claim 20.

The Nguyen patent further describes the use of a state machine class and a status object class. The status object class is updated if a transition state associated with an evaluation function linked to the state machine class is satisfied. Therefore, the status object class can be used to identify a state of the state machine class. (See generally, column 2, lines 24-40.) This teaching of the Nguyen patent does not approach the limitation of claim 20 that recites: "applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object." More specifically, there is nothing in the Nguyen patent that indicates that the status object class is *applied to* the state machine class to *update* the state of the state machine class, as recited in claim 20. At most, the status object class in Nguyen *monitors* the state machine class to identity when the state machine class undergoes a state change. Monitoring the state machine in Nguyen is not the same as actively participating in causing the state machine to change states, as recited in claim 20. And, in addition, the state machine class in Nguyen changes its state on its own. There is no compelling reason to think that Nguyen would use the status object class to update a state of the state machine class.

Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-discussed limitations of claim 20. This portion of Nguyen describes the use of an action 130 designed to implement an action when a state of the state machine class changes. This teaching has nothing to do with "applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object", as recited in claim 20.

In accordance with the above, Applicant submits that the Nguyen patent does not cure the deficiencies of the Feridum patent. Accordingly, at least claim 20 is not rendered obvious by the Feridum and Nguyen combination.

Claim 28 recites:

> An apparatus comprising:
> a plurality of event consumers; and
> an event correlator coupled to the plurality of event consumers, the event correlator to receive events from at least one event source and to receive data elements from at least one data source, the event correlator further to receive at least one correlation function configured to correlate events and data elements and to apply the received events and the received data elements to the correlation function, wherein the correlation function is implemented by a state machine having an associated update consumer that updates the state of the state machine, and wherein the event correlator generates a specific event if the received events and the received data satisfy the correlation function.

Claim 28 includes the limitation "wherein the correlation function is implemented by a state machine having an associated update consumer that updates the state of the state machine." For the same reasons presented in connection with the response to the rejection of claim 20, Applicant submits at least claim 28 is not rendered obvious by the Feridum and Nguyen combination.

Claim 35 recites:

> One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:
> receive a plurality of events;
> identify a plurality of correlation functions configured to correlate the plurality of events, wherein each of the plurality of correlation functions is implemented as a state machine having an associated update consumer;
> apply the plurality of events to the plurality of correlation functions to determine whether any of the plurality of correlation functions are satisfied by the plurality of events, wherein the plurality of events are applied by causing update consumers associated with each event to update the state of the associated state machine; and

generate a specific event if one of the plurality of correlation functions is satisfied by the plurality of events.

Claim 35 includes the limitation "wherein the plurality of events are applied by causing update consumers associated with each event to update the state of the associated state machine." For the same reasons presented in connection with the response to the rejection of claims 20 and 28, Applicant submits at least claim 35 is not rendered obvious by the Feridum and Nguyen combination.

Claim 40 recites:

> A computer-implemented method comprising:
> receiving events from event providers;
> creating a first state machine;
> creating a second state machine;
> associating a first event type with the first state machine, wherein the first state machine has an associated first update consumer to update the state of the first state machine;
> associating a second event type with the second state machine, wherein the second state machine has an associated second update consumer to update the state of the second state machine;
> in response to receiving an event having a first event type, applying the first update consumer to the first state machine;
> in response to receiving an event having a second event type, applying the second update consumer to the second state machine; and
> if the events are correlated:
> generating an additional event; and
> sending the additional event to an event consumer.

The Feridum patent describes the use of correlation rules implemented as state machines to recognize patterns of one or more events indicative of given conditions. According to Feridum, the observed events are examined to determine if one or more defined event patterns has occurred. If an event pattern is recognized, an event correlator may be used to take a defined action. (See Feridum, column 2, lines 42-62.)

The Feridum patent does not teach or suggest at least "wherein the first state machine has an associated first update consumer to update the state of the first state machine;... wherein the second state machine has an associated second update consumer to update the state of the second state machine;... in response to receiving an event having a first event type, applying the first update consumer to the first state machine; [and] in response to receiving an event having a second event type, applying the second update consumer to the second state machine", as recited in claim 40. The Office realizes that the Feridum patent doest not teach at least these limitations of claim 40 and attempts to cure this by combining Feridum with Nguyen.

The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-40, and column 6, lines 26-35 of Nguyen teaches the indicated limitations of claim 40. The Applicant disagrees for the following reasons.

Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen describes a status object that includes current information related to a state of a monitored system component. The status object receives information pertaining to the monitored system component and transitions its state if a transition condition associated with the status object is satisfied. If this occurs, some action is taken by the status object. (See generally, column 2, lines 1-14.) There is nothing in this section of the Nguyen patent that teaches or suggests the limitations from claim 40 that include: "wherein the first state machine has an associated first update consumer to update the state of the first state machine;... wherein the second state machine has an associated second update consumer to update the state of the second state machine;... in response to receiving an event having a first event type, applying the first update consumer to the first state machine; [and] in response to

receiving an event having a second event type, applying the second update consumer to the second state machine."

The Nguyen patent further describes the use of a state machine class and a status object class. The status object class is updated if a transition state associated with an evaluation function linked to the state machine class is satisfied. Therefore, the status object class can be used to identify a state of the state machine class. (See generally, column 2, lines 24-40.) This teaching of the Nguyen patent does not approach the limitations of claim 40 that recite: "wherein the first state machine has an associated first update consumer to update the state of the first state machine;... wherein the second state machine has an associated second update consumer to update the state of the second state machine;... in response to receiving an event having a first event type, applying the first update consumer to the first state machine; [and] in response to receiving an event having a second event type, applying the second update consumer to the second state machine."

More specifically, there is nothing in the Nguyen patent that indicates that the status object class *updates* the state of the state machine class, as recited in claim 40. At most, the status object class in Nguyen *monitors* the state machine class to identity when the state machine class undergoes a state change. Monitoring the state machine in Nguyen is not the same as actively participating in causing the state machine to change states, as recited in claim 40. And, in addition, the state machine class changes its state on its own. Therefore, there is no compelling reason to think that Nguyen would use the status object class to update a state of the state machine class.

Column 6, lines 26-40 fails to teach or suggest the above-discussed limitations of claim 40. This portion of Nguyen describes the use of an action 130 designed to implement an action when a state of the state machine class changes. This teaching has nothing to do with "wherein the first state machine has an associated first update consumer to update the state of the first state machine;... wherein the second state machine has an associated second update consumer to update the state of the second state machine;... in response to receiving an event having a first event type, applying the first update consumer to the first state machine; [and] in response to receiving an event having a second event type, applying the second update consumer to the second state machine", as recited in claim 40.

In accordance with the above, Applicant submits that the Nguyen patent does not cure the deficiencies of the Feridum patent. Accordingly, at least claim 40 is not rendered obvious by the Feridum and Nguyen combination.

For the reasons presented above, Applicant respectfully submits claims 1, 11, 20, 28, 35 and 40 are at least allowable over the Feridum and Nguyen combination. The remaining dependent claims are allowable by virtue of their dependency on one of the discussed independent claims.

Conclusion

The pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of the subject application. If any issue remains unresolved that would prevent allowance of this case, **the Examiner is requested to urgently contact the undersigned attorney to resolve the issue**.

Respectfully Submitted,

Date: February 22, 2006

By: Tim R. Wyckoff

Tim R. Wyckoff
Lee & Hayes, pllc
Reg. No. 46,175
(206) 315-4001 ext. 110